eGovernment Initiatives

Integrated Acquisition Environment (IAE)

Summary XML Guidance

Version 1.1

February 2, 2004

“Unify and Simplify
Through Shared Services”

Prepared by
GSA eGOV Program Management Office
U.S. General Services Administration
1 Crystal Park, Mail Stop 1AA
2011 Crystal Drive, Suite 911
Arlington, VA 22202

Table of Contents

Purpose

According to the CIO Council, "The global economy is increasingly dependent upon creative solutions to escalating information technology demands. The potential advantages of Internet transmission of payloads of information have highlighted the need to integrate data across applications, systems, and organizations. With the Internet -- and particularly web-enabled exchange of data -- still in its relative infancy, we have a unique opportunity for governments, business, and industry to foster the cooperative international development of a standardized platform-independent syntax designed to facilitate identification, exchange, and display of data using web transfer protocols. Extensible Markup Language (XML) is emerging as the preeminent tool for developers interested in maximizing system interoperability." [CIOC XDG, Executive Summary][1]

The purpose of this document is to provide ***summary guidance*** concerning the use of XML technology for the IAE eGov initiative. The goal is to highlight the key points from the major references and Web sites that XML developers should consult for federal guidance and governance. Since this document is only a high level summary, it is assumed that, whenever necessary, developers will refer to the sources listed in the Key Federal XML References and web Sites section for more authoritative and definitive information and governance.

For its governance and details, this document refers to specific sections of other federal guidelines that are very likely to change in the near future. We will therefore update our guidance accordingly. However, we believe most of the guidance given here is unlikely to change in any substantial manner, although the section number references and details may differ.

Note that one of the efforts of IAE is to define a ***standard vocabulary*** for procurement-related transactions based upon our on-going examination of the acquisition process and interactions among IAE shared systems. We are examining both message broker and Web services architectures. Our standard data elements for transactions will eventually become part of an IAE-wide repository and eventually part of a federal XML registry.

Therefore, those systems that have not yet begun their XML development or are just beginning such development **MUST**[2] use this guidance and **MUST** use the standard vocabulary that IAE is developing with the help of our Subject Matter Experts (SMEs). Furthermore, those systems that have *already* developed XML Schema and/or Web services **SHOULD** use these guidelines to develop and define an interface that maps their XML to the standard vocabulary that IAE establishes.

NOTE: *When the federal guidelines mature from "Draft" to "Approved" status, all IAE systems will be* **REQUIRED** *to comply with this guidance.*

[1] For specification sources of citations, see XML Guidance Documents.

[2] The words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this guidance are to be interpreted as described in Internet Engineering Task Force (IETF) Request For Comments (RFC) 2119.

Your points of contact for this document are:

Teresa.Sorrenti@gsa.gov	IAE Project Manager
Earl.Warrington@gsa.gov	IAE Deputy Project Manager
Thomas.Peters@gsa.gov	IAE Standard eTransactions Lead
Kenneth.Sall@gsa.gov	Technical Architect / XML Specialist GSA eGov Program Management Office

IMPORTANT: At this time, a number of areas are still to be determined, as identified by "TBD". This document will be revised as frequently as any of the government guidance documents on which it is based is updated. For several issues that are currently being debated, see the "Unresolved Issues" section of this document.

Guidance Areas

1. Selecting XML Standards for Project Use

In general, developers should use XML technology that is based on World Wide Web Consortium (W3C) Recommendations, which is the most mature level of W3C specifications. See http://www.w3.org/TR/#Recommendations . The W3C Process Document describes the standards acceptance track; see http://www.w3.org/2003/06/Process-20030618/tr . In cases where there is a need for developers to use less mature W3C technology (i.e., Proposed Recommendations, Candidate Recommendations, Last Call Working Drafts, Working Drafts, or Notes), developers **SHALL** contact the IAE Program Manager for approval, which will be reviewed on a case by case basis. In general, any specification with a Working Draft status is almost certain to change in incompatible ways. Therefore, development based on Working Drafts is likely to be judged too risky for anything but demos, pilots, and proof-of-concepts.

Exceptions to this guidance follow:

- The status of a given development effort influences its need for mature specifications. For example, demos and pilots can be freer in this respect than can production systems which generally need to employ only technology that has reached Recommendation status.
- *SOAP 1.1 and WSDL 1.1* – Although SOAP 1.2 reached Recommendation status in June 2003, vendors may not yet support this version. Therefore, use of SOAP 1.1 and WSDL 1.1 is acceptable at this time.
- *SAX 1.0 and SAX 2.0* – Simple API for XML is a non-W3C de facto standard supported by nearly all modern XML parsers.
- Compliance with standards from recognized standards organizations such as ISO, IEEE, UN/CEFACT, ANSI, OASIS, OMG, OAG, etc. is strongly encouraged.

References: [CIOC XDG, Section 2.1] and [DON XDG, Section 6]

2. Importance of International Standards

It is important to realize IAE's need for alignment with international efforts. Our trading partners operate in a global economy; they as well as their software providers are likely to follow international standards, where applicable. Data exchange involves vendors from many countries, so international standards should be considered applicable. Vocabulary definition is largely a data modeling effort using Unified Modeling Language (UML), ISO/IEC 11179 data element naming, and Universal Business Language (UBL) principles, followed by XML Schema development to define the precise structure of message payloads. The use of proprietary specifications MUST be avoided.

"In recognition of XML's significance as a web-based end-to-end solution for information interchange, the Architecture and Infrastructure (AIT) committee of the U.S. Federal CIO Council created the Federal XML Working Group in June 2000. The Group's primary responsibilities are to partner with *national and international standards organizations* in the development of XML and to facilitate the U.S. Government's transition to XML for electronic data interchange (EDI)." [CIOC XDG, Executive Summary, italics added]

Specifically, in an effort to standardize across the procurement environment, IAE strongly endorses the use of the following emerging international standards:

- ISO/IEC 11179 naming conventions
- UN/CEFACT Core Components Technical Specification
- Universal Business Language (UBL) methodology

References: [CIOC XDG, Section 2.1 and Executive Summary] and [DON XDG, Section 6]

3. Creating ISO 11179 Names

The description that follows is a considerable simplification of a more involved process. As such, developers are directed to ISO and UBL documentation (see References) for the details.

"ISO 11179 part 5 [see references] provides a standard for creating data elements. This standard employs a dot notation and white space to separate the various parts of the element and multiple words in a part respectively. In order to meet XML requirements for component naming, the ISO 11179 name must be converted to a [XML] *Name Token*[3]. The ISO 11179 part 5 standard provides a way to precisely create a data element definition and name. Using or referencing this name in a schema provides analysts with a better understanding of XML component semantics, while using business terms as element names improves readability. Requiring types to conform to ISO 11179 conventions will facilitate automated analysis of schema components during any harmonization efforts. The upper and lower camel case conventions are adopted from ebXML." [DON XDG, Section 7.1.3.2]

An ISO 11179- compliant data element name consists of three parts:

[3] An XML Name Token is a string that begins with a letter and contains only certain characters. In particular, spaces and underscores are not permitted in a Name Token.

- An *"Object Class"* term, which describes the kind of thing to which you refer. This Object Class may consist of one or more words, some of which may be context terms. For example, the ISO 11179 name `'Acoustic Signal. Frequency. Measure'` has the Object Class `'Acoustic Signal'`. As another example, consider that the ISO 11179 name `'EFT_ Payment. Authorization. Date'` has the Object Class `'Payment'`.
- A *"Property Term"* which is the property of the thing to which you refer, which may consist of one or more words. For example, the ISO 11179 name `'Acoustic Signal. Frequency. Measure'` has the Property Term `'Frequency'`. Also, the ISO 11179 name `'EFT_ Payment. Authorization. Date'` has the Property Term `'Authorization'`.
- A *"Representation Term"* which identifies allowable values for an element. This list is taken from an enumerated list of allowable representation types (see below). For example, the ISO 11179 name `'Acoustic Signal. Frequency. Measure'` has the Representation Term `'Measure'`. Another example is the ISO 11179 name `'EFT_ Payment. Authorization. Date'` that has the Representation Term `'Date'`.
- Each term may have an additional *qualifier* term (signified by a trailing underscore) to provide additional information. In the example above, the Object Class has the qualifier `'EFT_'` (as in "Electronic Funds Transfer").
- Other examples of ISO 11179 names include `Product. NAICS. Code` and `Organization. DUNS. Identifier`, neither of which have qualifiers.

ISO 11179 **Data Element Name**	**Qualifier** **(optional)**	**Object Class**	**Property**	**Representation Type**	**XML Element Name**
Acoustic Signal. Frequency. Measure	[none]	Acoustic Signal	Frequency	Measure	AcousticFrequency
EFT_ Payment. Authorization. Date	EFT_	Payment	Authorization	Date	EFTPayentAuthorizationDate
Product. NAICS. Code	[none]	Product	NAICS	Code	ProductNAICSCode
Organization. DUNS. Identifier	[none]	Organization	DUNS	Identifier	OrganizationDUNSID

"XML components **MAY** be named after ISO 11179 data element names:

- XML Elements **SHOULD** be named after ISO 11179 data element definitions *when business terms do not exist.*
- XML Attributes **SHOULD** be named after ISO 11179 data elements.
- *XML Schema data types* **MUST** be named after ISO 11179 data elements."

Note the distinction that XSD (XML Schema) data types **MUST** be based on ISO 11179 names, whereas XML elements **MAY** be based on more familiar business terms. This means that instance documents (e.g., messages) may use names based on business terms.

The XML Schema data type based on the first example above would be `AcousticSignalFrequencyMeasure`. The XML element name would be `<AcousticFrequency>`, assuming this was the common business term. Similarly, the other examples result in the XML element names `EFTPaymentAuthorizationDate`, `ProductNAICSCode`, and `OrganizationDUNSID`.

As another example, the ISO 11179 name `"Address. City. Text"` results from the Object Class `"Address"`, the Property Term Qualifier `"City"`, the Property Term Noun `"Name"`, and the Representation Type `"Text"`. The XML Schema type would be `"AddressCityName"` and the XML element name would be `<AddressCity>`.

The allowable Representation Types (known as Core Component Types in UBL) are:

- Amount
- Binary Object (plus Graphic, Picture, Sound and Video)
- Code
- Date Time (plus Date and Time)
- Identifier
- Indicator
- Measure
- Numeric (plus Value, Rate and Percent)
- Quantity
- Text (plus Name)

Those shown in parentheses are known as Secondary Representation Types.

References: [CIOC XDG, Section 3.3.2] and [DON XDG, Section 7.1.3.1 and 7.1.3.2]

4. Creating XML Element Names from Business Terms

"The creation and/or selection of XML component names and business terms:
MUST involve domain subject matter experts (operational personnel, program managers, etc), functional data experts (database administrators, functional data manager, data modelers, etc…) and software developers. Application developers **MUST NOT** be left on their own to perform this function." [DON XDG, Section 7.1.3.3]

"Developers **SHOULD** use business terms instead of ISO 11179 compliant names for element names when appropriate business terms exist; however, the underlying ISO 11179 name **SHOULD** be captured….The ebXML deliverables define the concept of a *Business Term*. Business terms are commonly recognized words that are more appropriately used as XML element names, rather than the often-esoteric ISO 11179 conventions. Business terms improve the readability of schemas and instances, while the ISO 11179 names provide more precise and structured semantics. Both are desirable when business and technical personnel are working together to define XML grammars for the exchange of business information by IT systems.

"This guidance may appear confusing because on one hand the creation of ISO 11179 names is recommended, but on the other, business terms are recommended for XML element names. The guidance is to define ISO 11179 standard names and capture those names through the use of the Schema "type" while retaining readability through using business terms as element names. Since the XML Schema is XML, those analysts interested in finding out, for instance, that `"National Stock Number"` is a business term for `"Federal Material Item. Identification. Details"` can look at the underlying type name of the `<NationalStockNumber>` tag." [DON XDG, Section 7.1.3.1]

References: [CIOC XDG, Section 3.3.2] and [DON XDG, Section 7.1.3.1 and 7.1.3.3]

Note: See Figure 1 for an overview of this naming process.

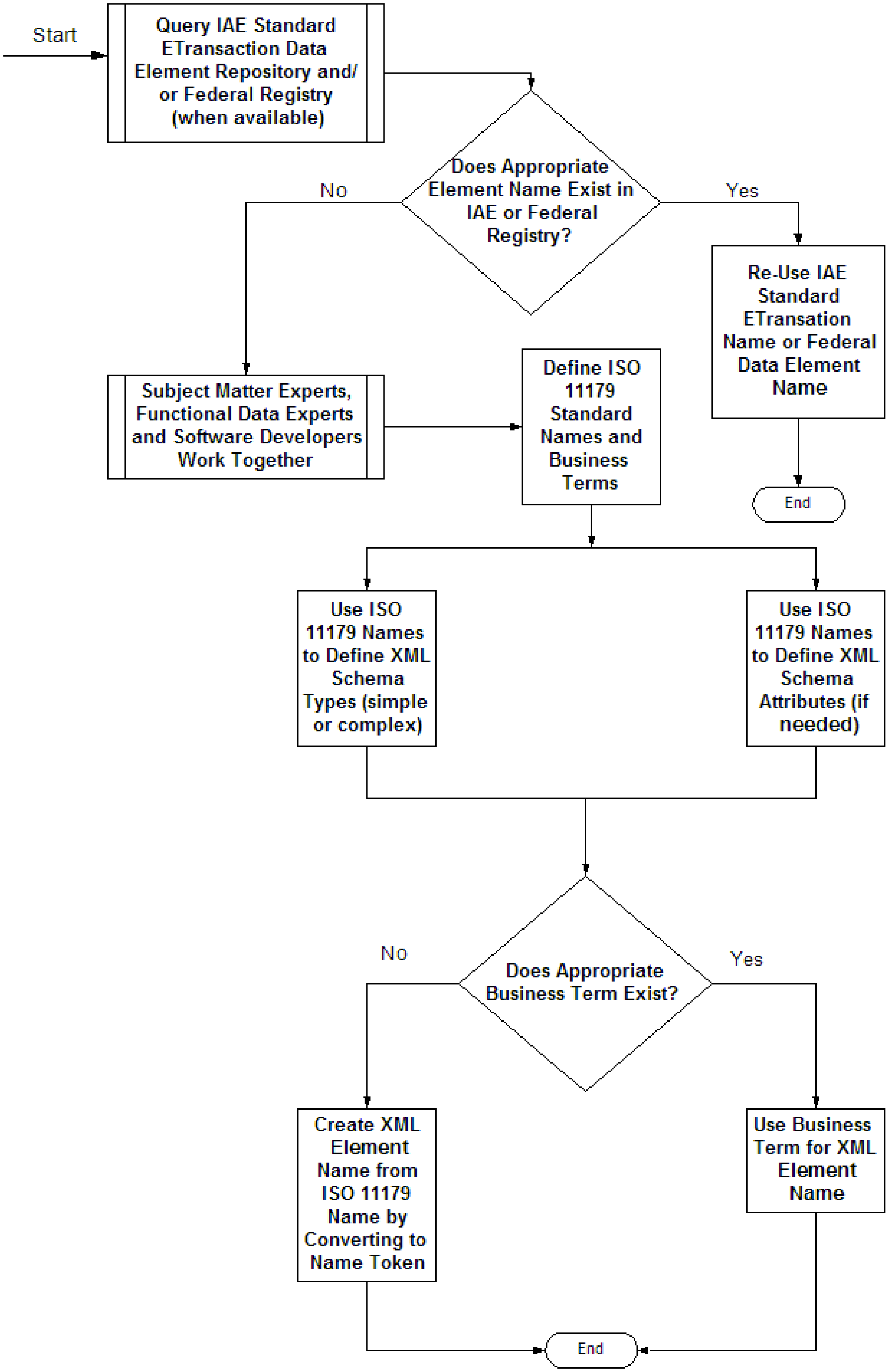
Start
Query IAE Standard ETransaction Data Element Repository and/ or Federal Registry (when available)
Does Appropriate Element Name Exist in IAE or Federal Registry?
No
Yes
Re-Use IAE Standard ETransation Name or Federal Data Element Name
End
Subject Matter Experts, Functional Data Experts and Software Developers Work Together
Define ISO 11179 Standard Names and Business Terms
Use ISO 11179 Names to Define XML Schema Types (simple or complex)
Use ISO 11179 Names to Define XML Schema Attributes (if needed)
Does Appropriate Business Term Exist?
No
Yes
Create XML Element Name from ISO 11179 Name by Converting to Name Token
Use Business Term for XML Element Name
End

Figure 1: XML Schema Type and XML Element Naming Process

5. Case Conventions

“Developers **SHALL** adopt the camel case convention, as defined by the ebXML Technical Architecture, when creating XML component names.

- XML Elements and XML Schema data types use *upper camel case*: The first letter in the name is upper case, as is the letter beginning each subsequent word.
- XML Attributes use *lower camel case*: Like upper camel case, except the first letter of the first word is lower case.” [DON XDG, Section 7.1.1]

Example:

```
<TotalBudgetAmount monetaryUnit=”USD”>12345678.90
</TotalBudgetAmount>
```

The convention the government has adopted is used in the business community in ebXML, UBL, and UN/CEFACT, as well as in the Department of Navy. The government’s intention to use this naming convention was announced in April 2002.

References: [CIOC XDG, Section 3.1] and [DON XDG, Section 7.1.1]

6. Usage of Acronyms and Abbreviations

The general guidance is to minimize the use of acronyms and abbreviations. Most XML standards organizations believe that abbreviations should be avoided and that acronyms should be used sparingly. However, in the government, this is quite difficult.

The following more specific guidance comes from the Dept. of Navy; the cited source provides an explanation for the reasoning behind the guidance.

- “Acronyms and abbreviations **SHOULD** generally be avoided in XML element and attribute names.
- For XML Schema data types, abbreviations **MUST** be avoided while acronyms MAY be used consistent with the rest of this guidance.
- When acronyms are used they MUST be in upper case. Abbreviations SHOULD be treated as words and expressed in upper camel case.
- While commonly used acronyms and abbreviations MAY be used in element and attribute names; the decision to use an acronym or abbreviation **SHALL** be made by program managers rather than by application developers. The decision to use an acronym or abbreviation **MUST** be based on the belief that its use will promote common understanding of the information both inside a community of interest as well as across multiple communities of interest. When an acronym or abbreviation does not come from a credible, identifiable source or when it introduces a margin for interpretation error, it **MUST NOT** be used.
- Acronyms and abbreviations used in component names **MUST** be spelled out in the component definition that is required to be included via schema annotations (as XML comments or inside XML Schema annotation `<xsd:documentation>` elements)….*References to authoritative sources from which the acronyms or abbreviations are taken SHOULD also be included in schema documentation*.” [DON XDG, Section 7.1.2; italics added]

References: [CIOC XDG, Section 3.2] and [DON XDG, Section 7.1.2]

7. Adding Comments and Metadata

“Developers **MUST** document XML element and XML Schema type definitions through XML comments, XML Schema annotations, a schema guide, or a data dictionary. These definitions **SHOULD** be related to underlying ISO 1179 data element definitions…. Mandating that schema developers take the time to provide element and Schema type definitions will facilitate identifying commonalities and reusable components.” [DON XDG, Section 7.2.3.2]

“Schemas and stylesheets **MUST** contain carefully thought out comments, reflecting basic information necessary to use and understand the documents. Generally, XML instances SHOULD NOT be documented.” [DON XDG, Section 7.3]

“A fully documented XML Schema may be quite verbose…A *compact* Schema that is equivalent to the *semantic* Schema may be quickly built for validation purposes.” [DON XDG, Section 7.2.3]

To promote interoperability, every schema and stylesheet **MUST** contain some basic *metadata*. The complete list appears in [DON XDG 7.3.2]. Illustrative metadata follows:

- Description of the purpose of the schema or stylesheet
- Developer point of contact information to include activity, name, and email. [May be waved by IAE PMO.]
- Change history log.

Schema, DTDs, and XSLT stylesheets should contain *version information* in the form of annotations, comments, or attributes, as appropriate per language syntax. See [DON XDG, Section 7.3.1.] for details.

References: [CIOC XDG, Section 4.3] and [DON XDG, Section 7.2.3 and 7.3]

Note: See also [DON XDG, Appendix F – *Sample XML Document Headers*.]

8. When to Use XML Schema vs. DTDs

The use of XML Schema is clearly preferred over DTDs, especially for new development. DTDs are almost certainly an inappropriate choice for data-oriented exchanges since data type validation cannot be built into the DTD. Federal guidance is to use XML Schema except for document-oriented applications, for which DTDs are acceptable. These guidelines prefer XML Schema to non-W3C specifications such as RELAX NG.[4]

[4] Since RELAX NG became an ISO standard (ISO/IEC 19757-2:2003) in December 2003, the IAE PMO is currently evaluating the appropriateness of permitting RELAX NG as an alternative to XML Schema. As of this writing, however, XML Schema is still preferred as per federal guidelines.

“Only W3C-recommended languages **SHALL** be used …. for describing documents. Specifically, the DTD and the W3C recommended XML Schema language **SHALL** be used. *All activities developing data-oriented schemas in DTD syntax **SHOULD** plan on migrating to XML Schemas*….XML developers **MAY** elect to use DTDs for markup of data that is strictly document-oriented (paragraph, chapter, appendix...); however, the XML Schema language is preferred.” [DON XDG, Section 7.2.1, italics added.]

References: [CIOC XDG, Section 4.1] and [DON XDG, Section 7.2.1 and Appendix H]

9. Schema Development Methodology

Refer to Figure 1. The following guidance is from [DON XDG, Section 7.2.2]:

- “XML developers **SHOULD** adopt the practice of developing schemas based on information exchange requirements identified via business process modeling. Information and process modeling and the XML schema creation process **SHOULD** be separate and distinct steps.
- Schema development **SHOULD** take place as a team effort with functional data experts, business experts, program managers, and IT specialists all involved. The DON XML WG also strongly encourages collaboration among activities developing schemas within related information domains.
- Conversely, schema development **SHOULD NOT** be solely the function of IT specialists. XML component names in general **SHOULD NOT** be taken directly from underlying relational database table and column names, unless the elements within that database have been named and created in accordance with a …. standard that represents concurrence by an entire Community of Interest (COI).”

“XML is an information-sharing meta-language that is inherently hierarchical, lending itself to be better represented via graphical modeling languages that allow capture of object relationships vice key/key-reference relationships of normalized entities. ….[A]ctivities interested in capitalizing on XML as an information exchange medium take the time to learn UML. UML is rapidly becoming the de facto industry standard for system requirements analysis and business process and information modeling as well as software design. It provides a common language that business experts, managers and IT specialists can use throughout all phases of a system’s implementation (requirements discovery, analysis, business rules and workflow documentation, software design, and deployment).” [DON XDG, Section 7.2.2]

References: [CIOC XDG, Section 3.3 and 4.2] and [DON XDG, Section 7.2.2]

Note: DON XDG, *Appendix E – Combined XML Schema Example* presents a complete example that illustrates much of the guidance proposed here and in the original documents.

10. When to Use Attributes vs. Elements

The debate of elements vs. attributes has been raging since the SGML days of the early 1990’s that pre-dated XML. The discussions captured on the following page may be useful, but keep in

mind that messages dated prior to May 2001 are based on the limitations of DTDs rather than what is possible with XML Schema.
http://www.oasis-open.org/cover/elementsAndAttrs.html

Disadvantages of attributes include:

- Attributes are white-space-normalized
- Attribute order is not significant (order undefined despite document order)
- Cannot have more than one attribute with the same name for a given element
- Attributes cannot contain any internal structure

Advantage of attributes include:

- Attributes are more compact than elements
- Attributes are useful for capturing metadata

“One of the key schema design decisions is whether to represent an information element as an XML element or attribute. Once an information element has been declared an attribute, it cannot be extended further; for this reason, the use of attributes is not encouraged.…The use of attributes **SHOULD** be carefully considered. Attributes, if used, **SHOULD** provide extra metadata required to better understand the business value of an element.” [DON XDG, Section 7.4]

For example, consider these two implementations of a person’s name – first, the attribute-centric version:

```
<PersonName first=”John” middle=”Q” last=”Public” />
```

and then the element-centric version:

```
<PersonName>
     <First>John</First>
     <Middle>Q</Middle>
     <Last>Public</Last>
</PersonName>
```

The attribute version is certainly more compact, but that is its only advantage in this example. The element version, on the other hand, has a fixed hierarchy with the three child elements always in the order `<First>`, `<Middle>`, and `<Last>`, an ordering which cannot be enforced in the attribute version. Furthermore, if for some reason we needed to repeat one of the child elements (e.g., to have two middle names), this is easy to change in the element version but impossible in the attribute version which inventing a new attribute, such as `secondMiddleName`, since no attribute can be repeated within a single element.

While it might seem desirable to use attributes for code lists because in DTDs an enumerated list can be validated by a DTD whereas element content cannot, with the advent of XML Schema, this is no longer a sufficient reason. Element content can be validated in XSD and `<xsd:enumeration>` can constrain code list values.

References: [CIOC XDG, Section 6.1] and [DON XDG, Section 7.4]

11. Global vs. Local Elements and Attributes

Whether XML Schema types should be declared as an immediate child of `<xsd:schema>` with global scope of the entire schema, or whether they should be permitted to be declared locally within a specific `complexType` that uses them is a fiercely debated topic among developers. Global datatypes are also known as "named types"; they can be associated with any element in a schema.

- Data-centric schemas MUST use simple datatypes to the maximum extent possible. (Simple datatypes allow for straight-forward specification of data requirements beyond what was possible with DTDs.)
- Data-centric schemas that employ complex datatypes MUST define the complex datatypes as global; they SHOULD NOT use local complex types.
- Data centric schemas MUST use global elements; they SHOULD NOT use local elements.
- Data-centric schemas MUST NOT use attributes in place of data elements. They MAY use attributes for metadata, however.

The rational for these decisions and many additional fine points is given in [EPA XDR]. See also the "Unresolved Issues" section of this document.

Reference: [EPA XDR, Chapter 2: Datatypes and Chapter 3: Elements and Attributes]

12. Enumeration of Data Values (Code Lists)

"XML schema developers **SHOULD** use XML Schemas to express enumeration constraints on XML element and attribute values, when such enumerated lists are of reasonable length and when code lists are considered stable (not likely to change frequently). The decision to explicitly enumerate in a schema **SHOULD** be made by program managers based on the resulting size of the schema, bandwidth availability, and validation requirements. Code lists, from which enumerations are taken, **SHOULD** be referenced by URI or other pointers so that analysts can look up code values." [DON XDG, Section 7.2.3.3]

Some developers feel that large lists of enumerated values should be stored in databases, external to the XML Schema. This is only acceptable if the XML Schema *clearly* documents from where the full set of valid values can be obtained. Even then, if the values are static, such as the names or abbreviations of all the states in the United States, it is preferable to embed the values directly in the XML Schema.

Exception: XML Schema processors may experience difficulty when there are numerous long code lists. Therefore, we recommend that code lists containing more than 256 codes NOT be implemented as `<xsd:enumeration>` and applications be responsible for validating long lists.

See also the "Unresolved Issues" section of this document.

References: [CIOC XDG, Section 4.3.3] and [DON XDG, Section 7.2.3.3]

13. Constraining Data Values

Data types should be as tightly constrained as is reasonable. If a schema defines data type constraints based mostly on limiting the number of characters, for example, this is generally far too liberal and fails to take advantage of the full power of XML Schema. For example, the definition of a phone number should be based on `<xsd:pattern>` in general.

Exception: In the early stages of schema development, lax or lenient definitions may actually be beneficial until the actual constraints can be defined.

References: [none]

14. XML Namespaces

Details concerning the use of XML Namespaces are presently undergoing changes. Please consider the following guidance as provisional, to be updated 1QFY04. See "XML Schema Namespace and Versioning Strategy for the National Environmental Information Exchange Network" [appendix to EPA XDR]. See also the "Unresolved Issues" section of this document.

Every XML Schema MUST specify at least one namespace in the `<xsd:schema>` root element. Data-centric schemas:

- MUST use namespaces.
- MUST use namespace qualification for all schema constructs. This implies that the `<xsd:schema>` contains `elementFormDefault="qualified".`
- MUST declare the W3C Schema namespace.
- MUST use namespace qualification for all W3C Schema constructs.
- SHOULD use "xsd" as a namespace prefix for all W3C Schema constructs.
- MUST use target namespaces.
- SHOULD use a multiple-namespace configuration. [see below]
- MUST NOT use default namespaces.
- MUST use namespace qualification for all attributes. This implies that the `<xsd:schema>` contains `attributeFormDefault="qualified".`

Data-centric XML instance documents (e.g., data exchange messages):

- SHOULD list the storage location of the schema where the XML instance document validates in the root element. (In other words, instances must use the `xsi:schemaLocation` attribute that points to the location of the schema against which they can be validated. This implies declaring xsi in the root element as well:

```
<RootElement
xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
xsi:schemaLocation=“URL_of_schema” >
```

- MUST use namespace qualification for all elements.

Specific IAE namespaces based on URLs (rather than URNs), subject to change, are:

- IAE XML Schemas SHOULD use a target namespace identifier of `"http://www.gsa.gov/eGov/IAE"` [TBD] an a namespace prefix of “IAE”.

- Partner XML Schemas SHOULD use a target namespace identifier of `"http://www.gsa.gov/eGov/IAE/`*`Partner`*`"` [TBD], where "Partner" is replaced by an acronym, such as FPDS, CCR, PPIRS, IGT, etc. The namespace prefix SHOULD be the acronym.

The EPA XML Design Rules document weighs the relative merits of using no XML namespace, a single namespace, or multiple namespaces. The authors conclude that for a large project, *multiple namespaces* are necessary: “Creating a ‘shared’ namespace in which commonly used XML constructs will be housed mitigates the risk of creating a lower level of interoperability. Allowing additional namespaces increases flexibility, and reduces cost and time by allowing organizational level namespaces. Limiting the number of additional possible namespaces mitigates the risks of the configuration becoming too complex and lowering interoperability. The additional complexity of multiple namespaces is minimal compared to the additional flexibility provided by the solution.” [EPA XDR, Appendix D]

Assuming that multiple namespaces are to be used, the EPA XML Design Rules identify three specific options:

- “Option 1: One network namespace and one namespace for each major data group
- Option 2: One network namespace, one namespace for each major data group, individual state environmental namespaces, and EPA media area namespaces
- Option 3: One network namespace, individual state environmental namespaces, and one EPA namespace.” [EPA XDR, Appendix D]

The “network namespace” is analogous to the IAE initiative namespace. EPA “data groups” correspond roughly to IAE systems such as FPDS, CCR, PPIRS, etc. IAE has no parallel to the EPA “individual state environmental namespaces”.

Option 1 discourages harmonization across data groups. On the other hand, using one namespace for each data group facilitates rapid development at different rates across groups, and ensures maximum flexibility in naming XML constructs, with the namespace providing the semantic context. “A separate namespace for each major data group creates interoperability problems. In the short term, the option will speed work because parties responsible for different areas will not have to make the data elements agree. In the long run, however, as documents are passed between more systems, additional mapping and translation of the same element in multiple namespaces will need to be harmonized. If the naming conventions prescribed by the XML design guidelines are used, creating unique names in different data groups should not be overly burdensome." [EPA XDR, Appendix D]

Option 2 doesn't apply to IAE since there is no need for state level namespaces.

Option 3 was the recommended choice for EPA: "Although this solution is not optimal for complete interoperability in future systems, it is a compromise solution that helps harmonize the frontend, ensuring interoperability by not segmenting the major data groups into their own namespaces. This option enables development to continue organizationally in the least time and at the least expense. It also gives states and EPA program offices greater flexibility as a solution for the organization and the Exchange Network." [EPA XDR, Appendix D]

However, since IAE is comprised of systems that may have already developed initial XML capabilities before there has been an opportunity to conduct the data normalization necessary to achieve agreement, we presently favor Option 1 strictly for practical reasons. This means that the IAE initiative will support both the elements `<IAE:Award>` and `<FPDS:Award>`, for example, even though they may not have the same structure. XSLT stylesheets can be used to map from one namespace to the other. As time permits, partner systems will work with the IAE PMO to reconcile element differences so that the need for a distinct `<FPDS:Award>` element will disappear and the cross-initiative `<IAE:Award>` element will prevail.

On the other hand, partner systems that have not yet begun development MUST follow the ISO 11179 and UBL methodology discussed throughout this document. They MUST use elements already defined in IAE or create new elements that are in harmony across the initiative.

References: [EPA XDR, Chapter 4: Namespaces and Appendix D]

15. Web Services

TBD: More specific guidance forthcoming.

The W3C defines a Web service as:

"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."
[http://www.w3.org/TR/2003/WD-ws-gloss-20030808/]

For the purposes of IAE, therefore, system stewards **SHALL NOT** refer to application interfaces as "Web services" unless they fit the above definition. In particular, do not use the terms "Web-enabled XML application" and "Web service" synomously since the latter is much more specific with respect to the underlying technology than the former. Simply exchanging XML over HTTP (or via FTP) without SOAP and potentially WSDL is a "Web-enabled" exchange, not a "Web service".

References: [W3C; anticipating guidance in DON XDR, Version 2.0]

16. Extensible Stylesheet Language Transformations (XSLT)

TBD: Guidance forthcoming.

References: [EPA XDR]

17. XML Software Tools

See http://www.xmlsoftware.com/ for a frequently updated, comprehensive XML software list by functional category.

18. Unresolved Issues

The IAE team and the GSA eGov PMO are aware of a number of issues that are still being debated in the federal space. A good source for details surrounding these controversial issues is the EPA document entitled "XML Design Rules and Conventions for the Environmental Exchange Network" [EPA XDR].

Issues that are being debated include:

- **Namespace strategies**, including definition of namespace URI for IAE and its systems:
  - ➢ Should the namespace URI be a URN or URL?
  - ➢ What should the exact namespace URI for IAE be? (The namespace prefix should be "IAE" but the exact URI or URN is unclear.)
  - ➢ At what point should systems such as FPDS, CCR, and PPIRS use their own namespace or immediately need to reconcile their elements with the common IAE namespace?
  - ➢ Should IAE elements immediately be promoted to a GSA namespace?
- **Global vs. local schema type definitions** – There are some experts who believe local type definitions are fine when reuse is not necessary and that promoting all types to a global level obscures which types are truly intended to be reusable. Others feel that all types should be global because sometimes the reusability of a type is not known when it is first defined; exposure increases the likelihood of reuse.
- **Code lists and enumerations** – Code lists SHOULD be accessible either directly via the schema as an inline enumeration or indirectly via `<xsd:import>` or `<xsd:include>`. Code lists maintained by different stewards SHOULD use separate namespaces. There is some controversy about how practical this guidance is both in terms of maintenance (inline lists) and in terms of performance (external lists). For example, should code lists reside only in databases with lax schema (not enumerating values), leaving code list validation to applications? While this is counter to the whole idea of

XML, in may be practical in some cases, especially if code lists are updated frequently and/or consist of numerous values. Appendix E: XML Enumeration and Code Lists of [EPA XDR] suggests a number of options:

- Enumerated list
- Instance extension
- QName in content
- Single type
- Multiple types
- Multiple namespace types – the recommended approach

"The OASIS Universal Business Language Technical Committee has defined a standard schema for code list modules. The current version of the schema module is in the process of becoming a UBL standard, after which it will be submitted as an OASIS and eventually an ISO standard." [EPA XDR, Appendix E, footnote 3]

"We recommend EPA and the states avoid enumerating lists of values within core schema. Instead, we recommend that all lists of values (other than simple Boolean such as yes/no, true/false, on/off) be created as schema modules. We further recommend EPA and the states adopt the multiple namespaced types option for defining code lists. We further recommend EPA and the states use the developing UBL schema for creating exchange network schema modules." [EPA XDR, Appendix E]

See also the OASIS UBL Code List Subcommittee**.**

- **Use of `<xsd:appInfo>`** - SHOULD NOT be used. The use of the `appinfo` element is considered very risky at this time because there is no guarantee that a given XML processor will properly pass the processing instructions to an application, or, if it does, that an application will be able to accept them or handle them properly.

Therefore, while the guidance summarized in this document represents the current best practices of experts, it is subject to change, especially in the areas listed above. Revisions of this document will be circulated each time an issue reaches resolution or a viewpoint is changed. While this may impact development, it is to your advantage to become familiar with these issues and perhaps to send comments to the contacts listed in the "Purpose" section above.

Reference: [EPA XDR, Appendices and document chapters]

Key Federal XML References and Web Sites

This section lists all of the major sources of XML Guidance in the federal space. If you encounter others that you believe are equally relevant, please contact Kenneth.Sall@gsa.gov .

Note: Several of the document references and site URLs have changed since version 1.0 of the "IAE Summary XML Guidance". Please obtain the latest copies using the links in this section. Notable changes include, but are not limited to:

- Core Components Technical Specification 2.01 [previously was version 2.0]
- UN/CEFACT Techniques and Methodologies site reorganization
- UBL 1.0 Beta [previously was version 0.7]
- Recommended XML Namespace for Government Organizations [previously was draft]
- DON XML link [old link is now password protected, so use new link]
- XML Design Rules and Conventions for the Environmental Exchange Network [previously was draft]
- Government Semantic XML Web Services Community of Practice [was Web Services Working Group]

1. XML Guidance Documents

Federal XML developers ***should adhere to*** the guidance in all three documents below. In the event that there is disagreement among the documents concerning certain points, follow the guidance as per the table below.

Guidance Lead	Areas Where Each Document Has Precedence for IAE
EPA XDR	Use of namespaces, global vs. local elements, code lists (enumerations), use of appInfo, file naming conventions, XSLT rules, detailed XML rules about schema and namespaces
DON XDG	schema design, IS0 11179 methodology, use of acronyms and abbreviations, case conventions, documentation (headers and per element)
CIOC XDG	Federal XML Registry; federal viewpoint
IAE Summary XML Guidance	IAE-specific interpretation of guidance from agencies, relevant specifications, attributes vs. elements

- CIO Council, XML Working Group, "XML Developer's Guide" [draft, dated April 2002]: http://xml.gov/documents/in_progress/developersguide.pdf **[CIOC XDG]**
- DON XML Working Group, "XML Developer's Guide", Version 1.1 [dated May 2002] and related material [See "Library" and then "Developer's Guide"; guide contains a very useful glossary; library includes a useful three-page XML Schema checklist] **[DON XDG]** https://quickplace.hq.navy.mil/quickplace/areatypes/navyxml/main.nsf
- EPA's "XML Design Rules and Conventions for the Environmental Exchange Network" [draft, dated April 2003]. This long and comprehensive document presents many XML

design issues, contains a helpful "Summary of XML Rules", and covers several issues that are unresolved as of this writing of our IAE guidance. **[EPA XDR]**

http://xml.gov/documents/completed/epa/EPAGuideSec1.pdf

http://xml.gov/documents/completed/epa/EPAGuideSec2.pdf

http://www.exchangenetwork.net/Documents/EP211S1_0_XML%20Design%20Rules%20Report.pdf [older version]

http://www.exchangenetwork.net/documents/sprsc_final.pdf

2. Related Guidance and Methodology

- "Summary of XML Rules" [September 2003; available as Appendix A of EPA XDR; very useful for those already familiar with XML Schema and XSLT details]
- "Recommended XML Namespace for Government Organizations" [dated August 2003]: http://xml.gov/documents/completed/lmi/GS301L1_namespace.pdf
- "XML Schema Namespace and Versioning Strategy for the National Environmental Information Exchange Network" [Dec. 2002; available as an appendix to EPA XDR]
- "UN/CEFACT Core Components Technical Specification", version 2.01 [November 2003]
  - UN/CEFACT - Core Component Technical Specification Version 2.01 (2003-11-15)
  - UN/CEFACT TMG - General Document Download Section [http://www.untmg.org/doc_tmg.html]
  - http://www.ebxml.org/specs/ [older versions of Core Components documents]
- "UN/CEFACT – Core Components User's Guide" [draft, October 2003]
  - Available from http://www.untmg.org/downloads/General/for_review/CC-UG-V20031030.zip
  - Primer that accompanies the Core Components Technical Specification.
  - Currently labeled "Not For Implementation" but still quite useful.
- ISO/IEC 11179, *Information technology -- Specification and standardization of data elements* Part 5: Naming and identification principles for data elements http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm [free version], especially ISO/IEC 11179-5:1995 http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=1758&ICS1=35&ICS2=40&ICS3= [purchased version]

  Note: Above documents are 1995 versions. An updated version is in development.
- "XML for e-Business" [July 2003; detailed tutorial on UBL and ISO 11179 by Eve Maler]

  http://www.ibiblio.org/bosak/ubl/tut/csw-xml-for-ebusiness.ppt
- Universal Business Language (UBL) – OASIS; available from:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

- UBL 1.0 Beta [1.0 final release is expected in February or March 2004] http://www.oasis-open.org/committees/ubl/lcsc/UBLv1-beta/download.zip
- On-line version of UBL 1.0 Beta

  http://www.oasis-open.org/committees/ubl/lcsc/UBLv1-beta/
- Unified Modeling Language (UML) - Rational

  http://www.rational.com/uml/index.jsp
- UMM: UN/CEFACT Modeling Methodology (based on UML)
  - UN/CEFACT - Modeling Methodology - Meta Model - Revision 12 (2003-01-17)
  - UN/CEFACT - UMM User Guide Revision 12 (2003-09-22)
- "XML Schema Design and Management Guide" [Hong Kong Special Administrative Region, November 2003], available from http://www.itsd.gov.hk/itsd/english/infra/eif.htm#if_spec
- "Naming and Design Rules for E-Government – The Danish Approach" [December 2003 paper from XML 2003 Conference] discusses strong vs. weak data typing for national vs. international transactions; available from http://www1.europarl.eu.int/forum/interop/dispatch.cgi/sibig/docProfile/100007/d20031218113107/No/20031211_XML2003_NDR_for_eGov_v21.ppt [reprinted with permission]

3. Related Web Sites

- Stay abreast of government guidelines and directions in XML technology. See these CIO Council web sites:
  - XML Working Group: http://xml.gov
  - Government Semantic XML Web Services Community of Practice: http://web-services.gov.

- Remain up-to-date with respect to adherence to the Federal Enterprise Architecture by monitoring the Web site www.FEAPMO.gov. This site was updated in August/September, 2003 with new FEA documents (Business Reference Model (BRM) version 2, Service Components Reference Model (SRM) 1.0 and Technical Reference Model (TRM) version 1.1). The Data and Information Reference Model (DRM) is due for public release in 2004.

- The government is actively investigating "E-Forms" as an offshoot of the (former) Web Services Working Group. Many vendors are participating in this effort, including Microsoft, Adobe, and PureEdge. If your client interface is mostly forms based, it may be wise to learn more about the government's E-Forms pilots. See http://web-services.gov and http://www.fenestra.com/eforms . The XForms approach of separating schema, presentation, and logic goes a long way toward enabling device-specific views of the same data. See http://www.w3.org/MarkUp/Forms/ for the W3C work in XForms. Several books about XForms can be ordered from Amazon.

- All W3C specifications, regardless of maturity, can be found at: http://www.w3.org/TR/

- A useful imagemap gateway to all major XML specifications is the "Big Picture of the XML Family of Specifications" by Ken Sall, located at: http://kensall.com/big-picture/

Glossary [5]

Term	Meaning
BRM	Business Reference Model; see FEAPMO
CIOC XDG	Chief Information Officers Council XML Developer's Guide
COI	Community of Interest
DOM	Document Object Model
DON XDG	Dept. of Navy XML Developer's Guide
DRM	Data and Information Reference Model; see FEAPMO
DTD	Document Type Definition
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
EPA XDR	Environmental Protection Agency XML Design Rules
FEAPMO	Federal Enterprise Architecture Program Management Office, which releases the various Reference Models (RMs), the source of governance for much of these guidelines; see PRM, BRM, SRM, DRM, and TRM
ISO/IEC 11179	International Organization for Standardization/International Electrotechnical Commission 11179 (data element naming and related standard methodology)
OASIS	Organization for the Advancement of Structured Information Standards
PRM	Performance Reference Model; see FEAPMO
RELAX NG	REgular LAnguage description for XML, Next Generation; clean, simple and powerful alternative to XML Schema from OASIS; pronounced "relax-ing".
SAX	Simple API for XML
SME	Subject Matter Expert [term also used in EDI for Small- and Medium-sized Enterprise]
SOAP	Simple Object Access Protocol [acronym is actually deprecated in SOAP 1.2]
SRM	Service Component Reference Model; see FEAPMO
TRM	Technical Reference Model; see FEAPMO
UBL	Universal Business Language
UDDI	Universal Discovery Description and Integration
UML	Unified Modeling Language
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema, also abbreviated as "WXS" (for W3C XML Schema)
XSLT	Extensible Stylesheet Language Transformations

[5] A more complete list of XML acronyms appears at http://kensall.com/big-picture/XML-acronyms.html with links to specifications. For even greater detail, see *XML Acronym Demystifier* at http://www.xml-acronym-demystifier.org which uses XML technology to query a collection of XML and Web Service related acronyms and abbreviations, and also returns a brief explanation and link to details.